

Universidad de Costa Rica

Escuela de Ciencias de la Computación e Informática

CI-0119 Proyecto Integrador de Arquitectura y Ensamblador

Profesor Jorge Castro Zeledón

Reporte de Errores

I Ciclo de 2020

Grupo 1 – Integrantes

- Gianfranco Bagnarello Hernández B70866
- Jose Alexis Valerio Ramírez B77863
- Kevin Guillermo Vargas Ramírez B57600
- Jey Caleb Ruiz Pinel B36237

I. Generador de Archivos

En el generador de datos para los sensores se implementaron 4 diferentes generadores de datos. Estos se utilizan para generar la lectura del sensor que se imprime en un ciclo. Inicialmente se genera e imprime la hora, que es de tamaño de 1 byte, luego los minutos, luego los segundos. Con estos se realizaron varias pruebas y generan valores adecuados. Para generar el valor de punto flotante, se utiliza la instrucción del sistema RDRAND, que genera un número random desde el hardware del procesador. Para que genere los valores en el rango adecuado, se le aplicó la fórmula matemática

$$(r = (eax \% (max-min)) + min)$$

Tal que eax es el registro de 32 bits donde RDRAND deposita el número random generado desde hardware, max es el límite superior y min es el inferior. Luego de ser procesado en la fórmula, el valor es transformado a punto flotante usando la instrucción CVTSI2SS. Para que el número nuevo tenga valores decimales, se le hace adicionalmente una división entre 1.02, que genera un valor muy parecido al original pero con varios decimales.

El ciclo que genera las lecturas anteriormente se repite 300 veces y las imprime en un archivo en formato binario.

II. Lector de Archivos

Varias funciones fueron implementadas en lenguaje ensamblador para la lectura del archivo generado por los sensores. Estas fueron `abrir(int * offset, char * nombre)` y `lector(char * fileDescriptor, int * hora, int * minuto, int * segundo, float * lectura)`. La función `abrir` abre el archivo con el nombre proporcionado en parámetros, y luego mueve el puntero del lector `offset bytes` para "brincarse" los bytes del encabezado, que se leen por medio de la función `scanf` en lenguaje C. Esta función retorna un file descriptor, que es un valor entero de 32 bits. Luego, en la función `lector`, se le pasa por parámetros ese file descriptor para que lea en orden la hora, minuto, segundo y lectura, y los lea en un ciclo que se repite 300 veces. Este ciclo guarda en arreglos de 300 valores cada uno de los datos leídos. Aquí se presenta el primer problema. A la hora de imprimir los valores guardados en estos arreglos, la función `scanf()` está generando un `segmentation fault (core dumped)`. El proceso de guardado no está generando problemas, sino más bien es el proceso de impresión. Al no saber si los valores guardados en esos arreglos son los correctos, se utilizó un lector hexadecimal para verificar. Los valores de las lecturas sí son horas minutos y segundos, pero el valor generado por `RDRAND` tiene problemas y está dando valores muy pequeños. Se modificó varias veces y se cambiaron los rangos mínimos y máximos para que funcionara pero sigue teniendo el mismo problema.

III. Interfaz Gráfica

La interfaz gráfica se implementó en el lenguaje de programación C++, usando el IDE de QtCreator. La interfaz no se logró conectar con el lector de datos por que no se sabe si los arreglos con los datos tienen valores erróneos. La interfaz tiene varios botones, donde llaman varias funciones de ensamblador como `maximo()`, `minimo()` y `promedio()`, que se implementaron y se probaron de manera independiente pero al no haber datos en los arreglos de lecturas, no se pueden utilizar, pero los botones funcionan. Además, la interfaz tiene dos modos, uno para mostrar las lecturas de los últimos 30 segundos, y otro para las lecturas de los últimos 5 minutos (es decir, de todo el archivo). Para motivos de probar la funcionalidad, se le ingresaron valores de prueba a la interfaz para que genere las líneas de estado para cada sensor y funciona adecuadamente.